

INPUTBOX KULLANIMI

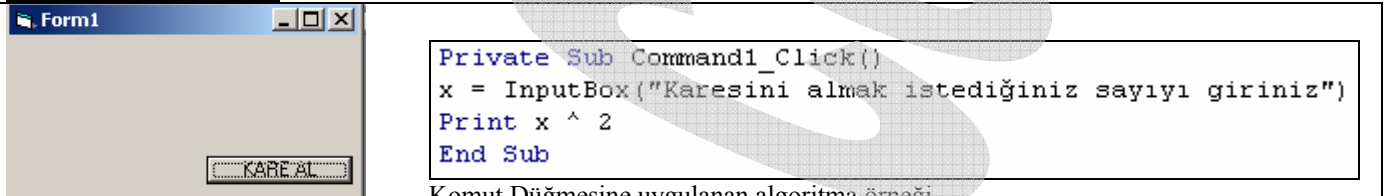
InputBox komutu, Basic programlama dilinde program algoritması içinde ekran aracılığı ile bir sorgu yönlendirerek, kullanıcının bilgi girmesini sağlamak amacıyla kullanılır. Bu komutun genel kullanım şekli (yazılış biçimi : syntax'ı) ;

Değişken = InputBox (“ Sorgu ile ilgili açıklama metni “)

Şeklinde. Bu komut diziliminde, parantez içinde belirtilen metin sorgu sırasında açılan InputBox penceresinde kullanıcıya bir diyalog olarak yansıtılır. “Değişken” olarak belirtilen ifade ise kullanıcı tarafından sorgu sırasında girilen verinin, aktarılacağı (girilen değeri üzerine alacak olan) değişkeni temsil etmektedir. Bu değişken, sayısal veya sayısal olmayan herhangi bir değişken olabilir.

Bu komutu, komut düğmesi tıklandığında ekrandan sorularak girilen bir sayının karesinin form üzerine yazdırılmasını amaçlayan bir program algoritması üzerinde uygulayalım.

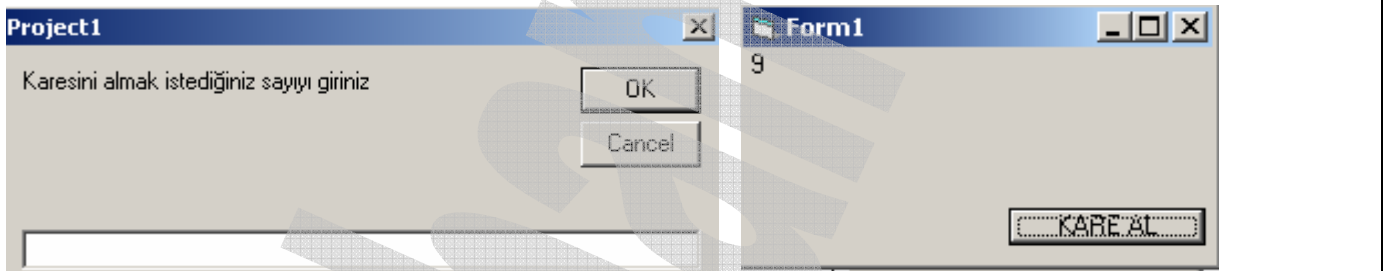
Örnek Uygulama :



```
Private Sub Command1_Click()  
x = InputBox("Karesini almak istediğiniz sayıyı giriniz")  
Print x ^ 2  
End Sub
```

Komut Düğmesine uygulanan algoritma örneği

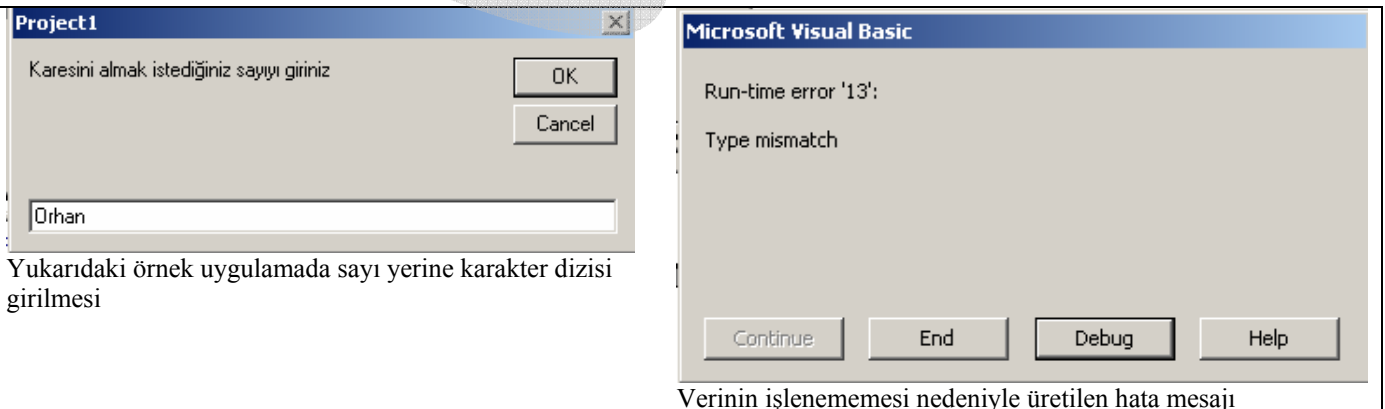
Form tasarım örneği



Form üzerindeki komut düğmesi tıklandığında aktif olan InputBox sorgu penceresi

Yandaki sorgu sırasında istenen değer 3 olarak girildiğinde form üzerine yazılan değer

InputBox komut dizisinde, eşitliğin solunda yer alan, üzerine değer alacak değişkenin türü ve buna bağlı olarak sorgu sırasında girilen değer (veya ifade) önem taşır. Eğer sayısal bir değişken tanımlanmışsa mutlaka girilecek değer **bir sayı** olmalıdır. Yukarıdaki örnekte sayısal bir değişken tanımlanmış ve sorgu sırasında girilen 3 değerinin karesi forma aktarılmıştır. Eğer, sayısal değişken için veri girişi istenmesi sırasında sayısal anlamı olmayan herhangi bir alfanümerik ifade girilirse, algoritma hata üretecektir. Örneğin ;



Yukarıdaki örnek uygulamada sayı yerine karakter dizisi girilmesi

Verinin işlenememesi nedeniyle üretilen hata mesajı

Eğer, kod yazılımı sırasında, tanımlanan değişken sayısal değil de **String** (sayısal olmayan = karakter dizisi = kelime = cümle) türü bir değişken ise, bu değişken, genel tanımlamalar sırasında string türü değişken olduğu tanımlanmalıdır. (Bu tür değişkenler değişken adı yanına \$ işareti konularak da belirtilebilir, x\$, Y\$, veri\$ gibi). Yukarıda kullanılan örneği, bir hal hatır sorma uygulamasında deneyelim;

```
Private Sub Command1_Click()  
X$ = InputBox("Görmeyeli ne var ne yok :)")  
Print X$  
End Sub
```

InputBox kullanarak Hal hatır sorma ifadesi ile ilgili algortima

Project1

Görmeyeli ne var ne yok :))

OK

Cancel

iyilik Sağlık abi

Sorgu sırasında karakter dizisi ifade girişi

Form1

iyilik Sağlık abi

Naaber

Form üzerinde elde edilen sonuç

Eşitlik tanımlamalarında, veya doğrudan yazdırma ifadelerinde karakter dizileri bilindiği üzere çift tırnak arasında tanımlanmaktadır: Bu örnekte ve yukarıdaki örnekte de sorgu sırasında ekrana yansıtılması istenen ifadeler InputBox komut diziliminde çift tırnak "" içine yazılmışlardır.

InputBox komut dizisinde, ekrana yansıtılmak istenen ifadeler de gerektiğinde, String değişkenlere aktarılarak da komut dizisi içinde kullanılabilir. Yukarıdaki örnekte kullanılan Hatır sorma ifadesini bu şekilde String değişken kullanarak komut dizisine aktarmak istersek aşağıdaki dizilimi uygulamamız gerekir.

```
Private Sub Command1_Click()  
A$ = "Görmeyeli ne var ne yok :)"  
X$ = InputBox(A$)  
Print X$  
End Sub
```

InputBox komut diziliminde açıklama ifadesinin değişken yardımıyla kullanılması

Project1

Görmeyeli ne var ne yok :))

OK

Cancel

Valla N'olsun, iyilik

Sorgu sırasında karakter dizisi ifade girişi

Form1


Valla N'olsun, iyilik

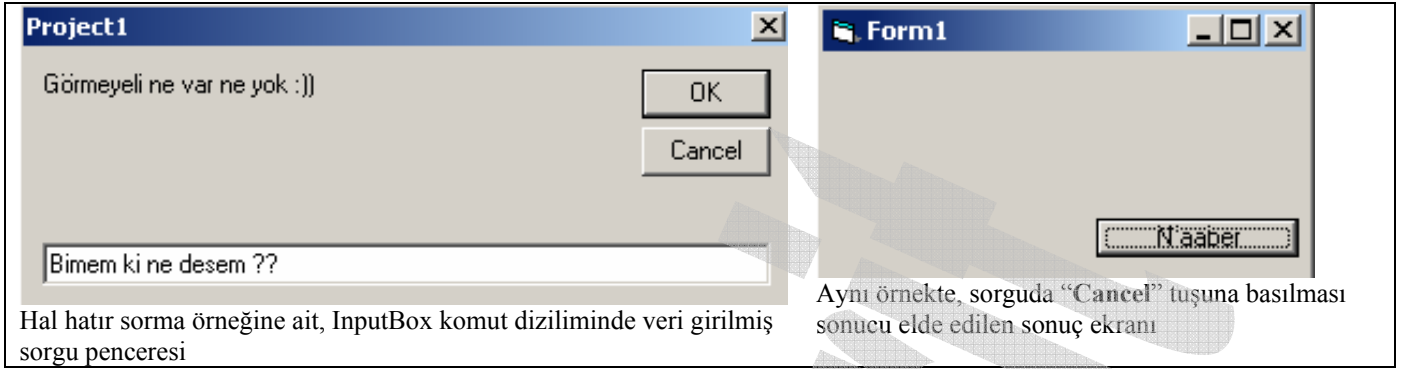
Naaber

Form üzerinde elde edilen sonuç

Burada InputBox komut diziliminde kullanılan; **X\$ = InputBox(A\$)** ifadesinde parantez içinde belirtilen A\$ değişkeni ayrıca çift tırnak ,ç,ne alınmaz. A\$ değişkeni program algoritmasında bir önceki satırda; **A\$ = "Görmeyeli ne var ne yok :))"** şeklinde tanımlanmıştır.

Buraya kadar anlatılan örneklerde, ister sayısal veri girişi ister String türü veri girişi olsun, InputBox komut diziliminin standart gösterim tarzında "OK" ve "Cancel" olmak üzere iki düğmenin yer aldığı gözlenmiş olmalıdır. Yukarıdaki örneklerde verilen sonuç ekranları daima veri girişinden sonra düğmesinin tıklanması ile oluşturulmuştur. Sorgu ekranı sırasında veri girişi sonucunda eğer bu örneklerde olduğu gibi "OK" düğmesi tıklanırsa girilen veri InputBox komut diziliminde tanımlanan değişkene değer olarak aktarılır ve girilen değere göre

işlem devam eder. Ancak veri girişinden sonra “OK” düğmesi yerine  tıklanırsa, değişkene değer aktarımı gerçekleşmez. İşlem bir sonraki satırdan devam eder.



Aynı durum sayısal veri ve sayısal değişken kullanımında da benzer şekilde gerçekleşir.

Eğer InputBox komut kullanımında değişken olarak String türü değişken kullanılmış ancak sayısal veriler kullanılmak isteniyorsa, girilen verileri sayısal anlamlara dönüştürmek için **Val** fonksiyonu kullanılır (Bakınız: OrhanCerit, VB ders notları)

Eğer sayısal bir değeri sayısal olmayan String türü bir değer dönüştürmek gerekirse bu işlem **STR\$** fonksiyonu ile gerçekleştirilir.

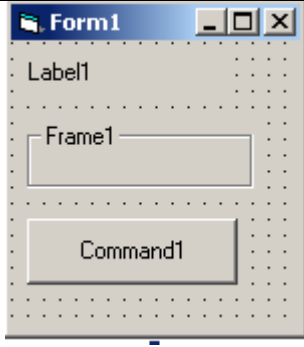
STR\$ fonksiyonunun kullanımı:

Bu fonksiyon; **değişken\$ = STR\$ (sayısal değişken)** veya **değişken\$ = STR\$ (sayı)** şeklinde kullanılır. Bu durumda fonksiyon parantez içindeki sayının veya sayısal değişkenin temsil ettiği değeri, sayısal anlamı olmayan karakter dizilimi şeklindeki ifadesini değişken\$ olarak tanımlana değişkene aktarır. Örneğin:

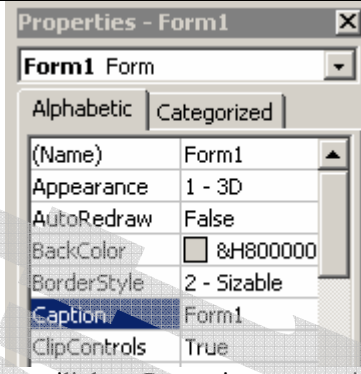
- A\$ = STR\$ (123) ifadesinde A\$'in değeri 1, 2 ve 3 karakterlerinden oluşan üç harfli bir kelimedir.
- A\$ = STR\$ (X) ifadesinde A\$'in değeri, X sayısal değişkeninde tanımlı olan sayısal değer karakter dizilimi olarak karşılığıdır..
- A\$ = “ALİ” + STR\$ (20) + “Yaşındadır” ifadesinde, A\$'in değeri; “ALİ20Yaşındadır” ifadesine eşit olmuştur.

Visual Basic Nesnelerinin Caption özellikleri

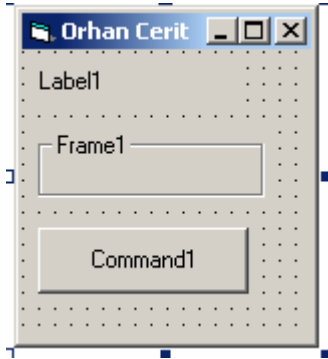
Visual Basic programlama dilinde, program tasarımında kullanılan pek çok nesnenin (Command button, Frame ve Label nesnelere gibi), program çalıştırıldığında (RUN edildiğinde) ekranda, bu nesnelere üzerinde görünen açıklama metinlerine “**Caption**” denilmektedir. Bu ekrana yansıtılan nesne başlıkları (caption) programcının tercihine göre çeşitli şekillerde değiştirilebilir. Bu değiştirme işleminin en basit yolu, seçilen nesnenin özelliklerini gösteren “**Properties**” penceresine “caption” kısmında yazan etiketi değiştirmektir.



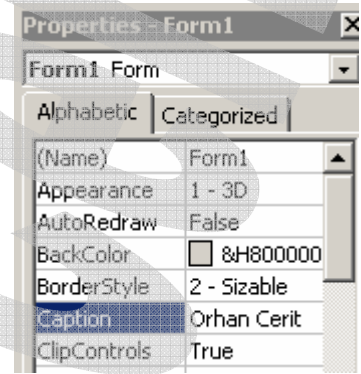
Form1 nesnesinin seçili olması durumunda, formun başlığı **Form1** olarak ötürmektedir



Form1 nesnesinin seçiliyken Properties penceresinde Caption özelliğinin değıştirilmeden önceki görünümü

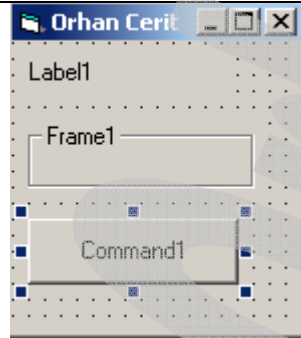


Aynı formun başlığının değıştirildikten sonraki görünümü

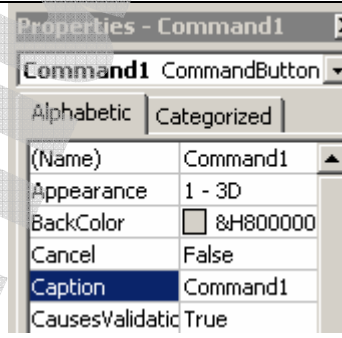


Form1 nesnesinin Properties penceresinde, Caption özelliğinin değıştirilmesi

Bu işlem diğeri nesnelere de seçilmesiyle, o nesnelere ait Properties (Özellikler) penceresinde benzer şekilde caption özelliğinin değıştirilerek de yapılabilir.

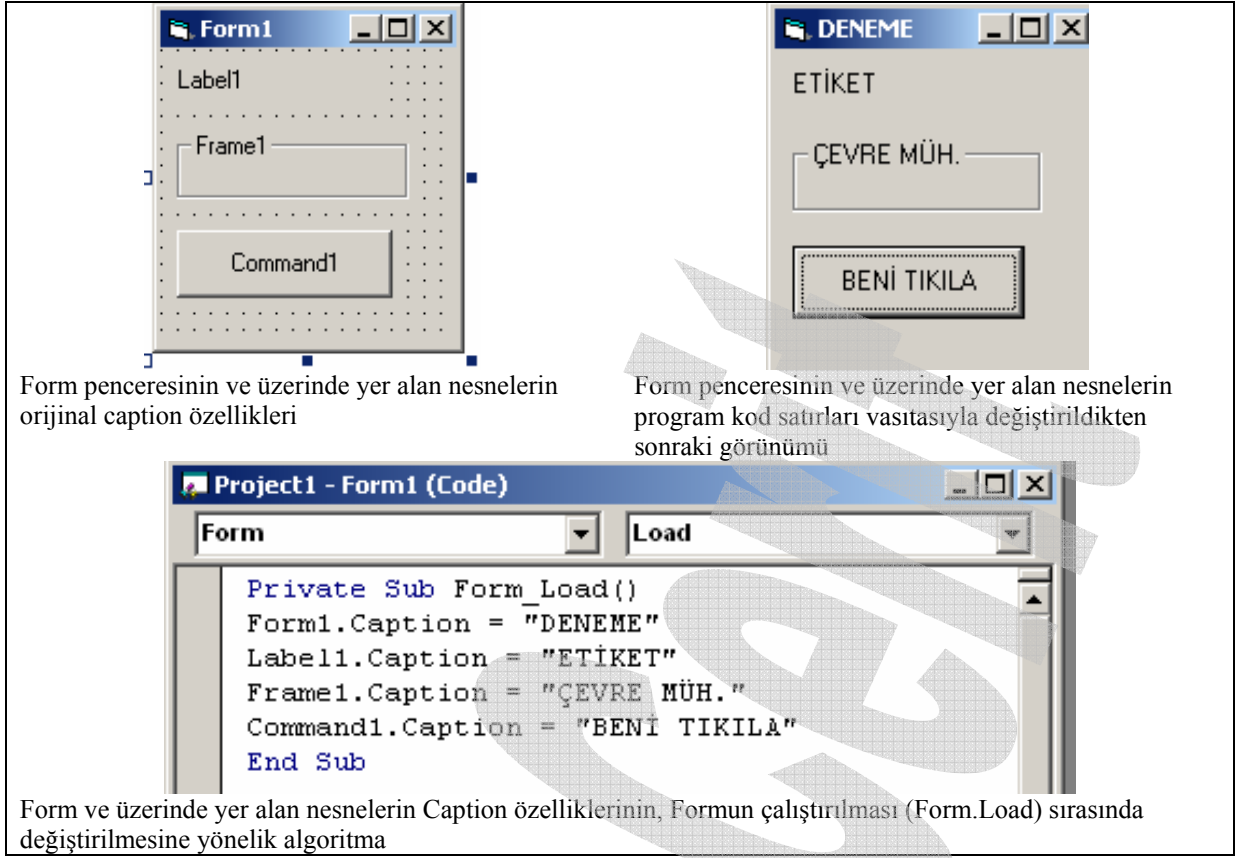


Form üzerindeki Command1 nesnesinin seçilmesi



Command1 nesnesi seçiliyken Caption özelliğinin değıştirilmesi

Söz konusu Caption özelliğinin, yukarıda ifade edildiği şekilde, Properties penceresinden yazılmak suretiyle değıştirilebileceği gibi, program algoritması içinde komut satırlarıyla bu özellik tanımlanarak da değıştirilebilir. Örneğin yukarıda ifade edilen form örneğinin ve üzerinde yer alan nesnelere başlıklarının (caption) programın çalıştırılması sırasında formun ilk yüklenmesi anında (Form_Load özelliğinin) komut satırlarının işlenmesi ile değıştirilmek istendiğini varsayalım.



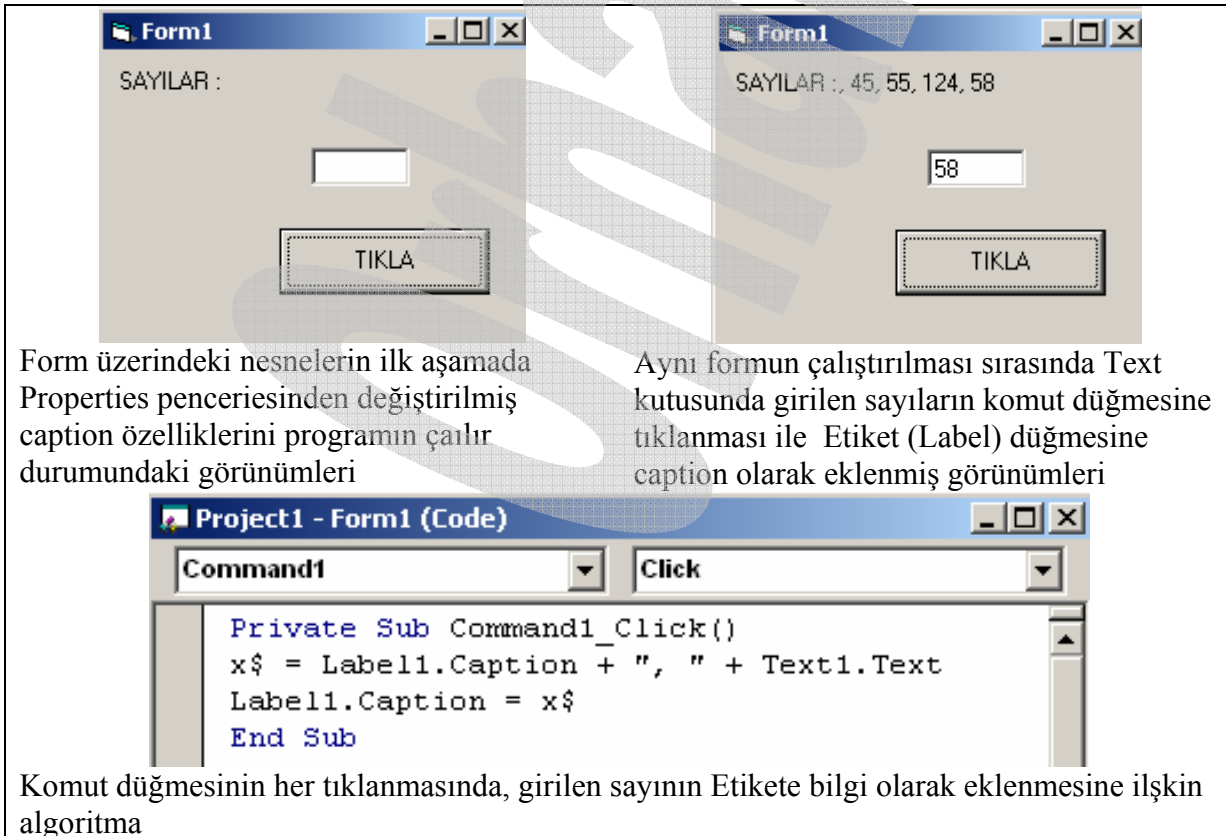
Form penceresinin ve üzerinde yer alan nesnelerin orijinal caption özellikleri

Form penceresinin ve üzerinde yer alan nesnelerin program kod satırları vasıtasıyla değiştirildikten sonraki görünümü

```
Private Sub Form_Load()  
Form1.Caption = "DENEME"  
Label1.Caption = "ETİKET"  
Frame1.Caption = "ÇEVRE MÜH."  
Command1.Caption = "BENİ TIKILA"  
End Sub
```

Form ve üzerinde yer alan nesnelerin Caption özelliklerinin, Formun çalıştırılması (Form.Load) sırasında değiştirilmesine yönelik algoritma

Form nesnelerinin Caption özelliklerinin bu şekilde program algoritması içinde değiştirilebilir olması, bu nesnelerin dinamik olarak, programın çalışması sırasında sürekli değiştirilmesiyle, istenen bilgilerin ekranda ard arda gösterilmesini sağlama amacıyla da kullanılmasına olanak verir. Örneğin Text kutusunda girilen verilerin komut düğmesine tıklanması ile Label (Etiket) kutusunda ardışıklı olarak yazdırılmak istendiğini var sayalım.



Form üzerindeki nesnelerin ilk aşamada Properties penceresinden değiştirilmiş caption özelliklerini programın çalışır durumundaki görünümleri

Aynı formun çalıştırılması sırasında Text kutusunda girilen sayıların komut düğmesine tıklanması ile Etiket (Label) düğmesine caption olarak eklenmiş görünümleri

```
Private Sub Command1_Click()  
x$ = Label1.Caption + ", " + Text1.Text  
Label1.Caption = x$  
End Sub
```

Komut düğmesinin her tıklanmasında, girilen sayının Etikete bilgi olarak eklenmesine ilişkin algoritma

Bu algorithmada;

x\$ = Label1.Caption + ", " + Text1.Text

komut satırı, komut düğmesi tıklandığında Label kutusu içinde mevcut bulunan başlığı , peşine bir virgül “,” ve Metin kutusunda girilmiş sayıyı ekleyerek **X\$** metin (String) değişkenine aktarmaktadır. Bu satırı, takip eden ;

Label1.Caption = x\$

Satırı da daha önce ifade edilen örneklerde olduğu gibi, X\$ dizi değişkeninde toplanmış olan metin ifadesini Label1 düğmesine etiket olarak atamaktadır.

Orhan Çamlı